



Universitat Autònoma
de Barcelona

Contabilizació mediante lector óptico

Memoria del proyecto
de Ingeniería Técnica en
Informàtica de Sistemas
realizado por
Jordi Malé Marbà
y dirigido por
Jordi Pons Aróategui

Escola d'Enginyeria
Sabadell, Junio de 2010

Jordi Pons Aróztegui,
profesor de la Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el trabajo al que corresponde la presente
memoria ha sido realizado bajo su dirección
por Jordi Malé Marbà

Y pera que conste firma la presente.
Sabadell, Junio de 2010

Firmado: Jordi Pons Aróztegui

Xavier Urtrasun Villanueva, de la empresa
de UNIT 4,

CERTIFICA:

Que el trabajo al que corresponde la presente
memoria ha sido realizado bajo su dirección
per Jordi Malé Marbà

Y pera que conste firma la presente.
Sabadell, Junio de 2010

Firmado: Xavier Urtrasun Villanueva

Resumen

Se ha desarrollado en la empresa UNIT 4 de acuerdo con un convenio que tienen con la Escuela d'Enginyeria de la UAB. Esta empresa se dedica al desarrollo de SW para pequeñas y grandes empresas y disponen de un ERP (ekon) que incluye una sección (ekon finanzas) destinada a la gestión de finanzas. En este caso el proyecto estará dentro del apartado de gestión de facturas.

La gestión de las facturas en muchos casos es un proceso totalmente manual o con un mínimo soporte informático, eso supone un gasto innecesario de tiempo y de personal. Automatizar este proceso puede suponer una mejora importante tanto en el ahorro de costes, como en la reducción de documentación y en la gestión de impagados. A través del lápiz contable se tiene que poder convertir una factura de papel en una factura pasada a PC. El hecho de tratar el documento de forma electrónica presenta innumerables ventajas para la empresa, ahorrando principalmente costes en gestión. Las aplicaciones informáticas, en este caso ekon finanzas, se tienen que adaptar al tipo de formato de factura para hacer llegar al cliente final los ahorros en gestión.

La tecnología con la que se realizara el proyecto es con Eclipse y el lenguaje de programación será Java además de las aplicación karat para crear los formularios y SQL server para la base de datos.

Tabla de contenido

1. Introducción..... - 3 -
1.1. Marco de trabajo..... - 3 -
1.1.1. Convenio UAB(EE)-UNIT 4..... - 3 -
1.1.2. La empresa - 3 -
1.2. Objetivos del proyecto..... - 4 -
1.3. Contenido de la memoria - 4 -
2. Planificación del proyecto - 5 -
2.1. Modelo de desarrollo..... - 5 -
2.2. Planificación inicial..... - 5 -
2.3. Desviación respecto la planificación - 7 -
3. Definición de requerimientos - 9 -
3.1. Sistema actual..... - 9 -
3.1.1. Descripción..... - 9 -
3.1.2. Diagnóstico..... - 9 -
3.2. Sistema propuesto..... - 9 -
3.2.1. Objetivos - 9 -
3.2.2. Objetos..... - 9 -
3.2.3. Reglas de gestión..... - 15 -
3.2.4. Funcionalidades..... - 16 -
3.2.5. Otros requerimientos no funcionales..... - 16 -
3.2.6. Requerimientos de Software - 17 -
3.2.7. Plan de validación - 17 -
3.3. Costes, beneficios y riesgos..... - 17 -
4. Análisis y diseño - 19 -
4.1. Introducción..... - 19 -
4.2. Contabilización mediante lector óptico - 19 -
4.2.1. Introducción de datos en Pantalla previa..... - 19 -
4.2.2. Introducción de datos en la entrada automática - 20 -
4.3. Modelo físico..... - 24 -
4.3.1. Diagrama de nuevas tablas - 24 -
4.3.2. Descripción de componentes..... - 24 -
4.4. Especificaciones de procesos..... - 26 -
4.4.1. Contabilización mediante lector óptico - 26 -
4.4.2. Datos iniciales - 26 -
4.4.3. Entrada Factura(Lector óptico) - 29 -
4.5. Plan de Pruebas - 36 -
4.5.1. Prueba 1 - 36 -
4.5.2. Prueba 2..... - 36 -
4.5.3. Prueba 3 - 37 -
4.5.4. Prueba 4..... - 38 -
4.5.5. Prueba 5..... - 38 -
5. Construcción..... - 41 -
5.1. Tecnología utilizada - 41 -

5.1.1. Lenguajes de programación.....	- 41 -
5.1.2. Entorno de desarrollo	- 41 -
5.1.3. Estilo de Codificación	- 42 -
5.2. Normas	- 42 -
5.2.1. Formulario	- 43 -
5.2.2. Entorno Eclipse	- 43 -
6. Pruebas	- 45 -
6.1. Pruebas unitarias.....	- 45 -
6.1.1. Prueba 1	- 45 -
6.1.2. Prueba 2	- 45 -
6.1.3. Prueba 3	- 46 -
6.1.4. Prueba 4	- 46 -
6.1.5. Prueba 5	- 46 -
7. Conclusiones	- 47 -
7.1. Objetivos cumplidos.....	- 47 -
7.2. Líneas de trabajo abiertas	- 47 -
7.3. Valoración personal.....	- 47 -

Tabla de imágenes

Figura 1: Planificación inicial proyecto - 6 -

Figura 2: Diagrama de Gant - 7 -

Figura 3: Formulario Pantalla Previa - 10 -

Figura 4: Pantalla Previa: Datos comunes a todas las facturas - 10 -

Figura 5: Pantalla Previa: Restricciones..... - 11 -

Figura 6: Pantalla Previa: Valores por defecto..... - 11 -

Figura 7 Formulario Entrada Automática - 12 -

Figura 8: Entrada Automática: Lectura de importes - 12 -

Figura 9: Entrada Automática: Entrada de datos..... - 13 -

Figura 10: Entrada Automática: Datos factura..... - 13 -

Figura 11: Entrada Automática: Desglose de IVA..... - 13 -

Figura 12: Entrada Automática: Desglose de IRPF - 14 -

Figura 13: Entrada Automática: Desglose de Vencimientos..... - 14 -

Figura 14: Entrada Automática: Desglose de Contrapartida..... - 14 -

Figura 15: Entrada Automática: Desglose analítico..... - 15 -

Figura 16: Entrada Automática: Asientos generados - 15 -

Figura 17: Pantalla Previa - 19 -

Figura 18 Entrada Automática - 21 -

Figura 19: Entrada Automática: Divisa..... - 21 -

Figura 20: Tabla bases de datos - 24 -

Figura 21: Pantalla Previa - 29 -

Figura 22: Entrada Automática - 34 -

Figura 23: Entrada Automática: Desglose IVA - 34 -

Figura 24: Entrada Automática: Desglose IRPF - 35 -

Figura 25: Entrada Automática: Desglose Vencimientos - 35 -

Figura 26: Entrada Automática: Desglose Contrapartida..... - 35 -

Figura 27: Entrada Automática: Desglose Analítico..... - 35 -

Figura 28: Entrada Automática: Asientos generados - 36 -

1. Introducción

1.1. Marco de trabajo

1.1.1. Convenio de colaboracion entre UNIT 4 y la UAB

Como se explica en el resumen, este proyecto se ha desarrollado dentro del marco de un convenio entre la Universitat Autònoma de Barcelona y la empresa Unit 4, en el cual, se estipulan 560 horas para realizar un trabajo de desarrollo de software. Mediante este convenio, el alumno puede adquirir conocimientos de cómo funciona una empresa de desarrollo de software de gestión y de tecnologías de la información, a la vez que este desarrollo sirve como trabajo de final de carrera.

1.1.2. La empresa

UNIT 4 tiene una experiencia durante más de cuarenta años en el mundo de la ingeniería de software. Posee un amplio abanico de soluciones de gestión de última generación lo cual la hace ser una de las primeras empresas españolas en tecnologías de la información y comunicaciones.

Actualmente, UNIT 4 cuenta con 500 empleados y posee cinco centros de trabajo ubicados en Barcelona, Madrid, Valencia, Donostia y Zaragoza, también cuenta con presencia internacional en países como Francia, Italia, Suiza, Grecia, Portugal y México.

Mediante la oferta de soluciones y objetivos, su misión es la de proporcionar una auténtica ventaja competitiva y diferenciadora a sus clientes mediante la implantación de soluciones informáticas de gestión.

Sus principales objetivos como empresa son:

- Mantener su posición de liderazgo como proveedor global de soluciones y servicios TIC.
- Estar por delante en investigación de nuevas tecnologías y desarrollo de nuevas soluciones.

El departamento en el cual se ha realizado el proyecto es el departamento de finanzas, que está ubicado dentro de la sección fábrica de la empresa. Este departamento es el encargado de realizar las aplicaciones que inciden en la contabilidad general, analítica y presupuestaria, así como en el seguimiento de los cobros y pagos mediante la gestión de cartera. La aplicación que se desarrolla tiene el nombre de *ekon finanzas*, la cual sus módulos principales son:

- *Ficheros contables*: módulo que se encarga del mantenimiento de entidades, libros de entidad, jerarquías, planes, etc.
- *Contabilidad general*: comprende la entrada de asientos, extractos de cuenta, listado de diarios, consulta de apuntes, etc.
- *Gestión de cartera*: módulo que posee balances de antigüedad, consultas de cartera pendiente, etc.
- *Informes contables*: módulo en el que se encuadrará el proyecto que es el encargado de obtener los distintos tipos de balances e informes contables.
- *Gestión de tesorería*: comprende a las entidades bancarias, cuentas bancarias, etc.

- *Tratamiento fiscal:* módulo que se encarga de la gestión del IVA y demás impuestos.

1.2. Objetivos del proyecto

El objetivo del proyecto es crear un nuevo proceso para el registro de facturas que permita entrar facturas tradicionales en papel sin necesidad de teclear nada, sólo pasando el lector óptico por encima de unos campos concretos de la factura tradicional de papel. Esto nos generará un borrador, gracias al cual se podrán crear asientos con facilidad sin requerir conocimientos avanzados de *ekon finanzas*.

Los beneficios que nos aportara la utilización del lápiz contable serán que nos permitirá entrar facturas con solo pasar el lápiz por encima de unos pocos campos de la factura y el formulario del ordenador se irá rellenando automáticamente partiendo de los campos leídos de una forma muy sencilla y sin ningún esfuerzo y con mucha rapidez.

1.3. Contenido de la memoria

Seguidamente, se describirán los puntos de los cuales constará la memoria del proyecto:

- **Introducción:** Apartado en el cual se explica de forma breve el marco de ubicación del proyecto, así como los objetivos marcados.
- **Planificación del proyecto:** En este capítulo se podrá observar la planificación inicial de las tareas a realizar en el proyecto con su duración, y su desviación final. También se explicará el método de desarrollo de software utilizado.
- **Definición de requerimientos:** Se hará una pequeña descripción del sistema actual, del sistema propuesto (objetos, funcionalidades, etc.), y de los beneficios y riesgos del proyecto.
- **Análisis y diseño:** Se profundizará en lo especificado en el capítulo anterior y se especificarán sus componentes en los dos niveles, así como de una descripción de los procesos involucrados. También se realizará un plan de pruebas a realizar una vez codificada la aplicación.
- **Construcción:** Apartado en el cual se describirá la tecnología utilizada en el desarrollo, así como el estilo de codificación usado.
- **Pruebas:** Se describen en este apartado todas las pruebas realizadas sobre la aplicación y el resultado de éstas para comprobar la eficacia del programa.
- **Conclusiones.** Se redactarán los objetivos conseguidos, las líneas de trabajo abiertas así como una valoración personal sobre el proyecto.
- **Bibliografía:** Detalle de recursos informativos utilizados para realizar el aplicativo.

2. Planificación del proyecto

2.1. Modelo de desarrollo

El proyecto se realizará utilizando una metodología de creación de software denominada modelo lineal, que tiene las siguientes características:

- Después de realizar una etapa se realizará la siguiente.
- Cuando estemos en una etapa intermedia, con este método se puede volver a una etapa anterior y realizar modificaciones si hubiera algún error.
- El desarrollo se hace por incrementos, esto significa que al haber acabado de realizar una funcionalidad se probará, lo cual hace que las pruebas sean más exhaustivas.

El principal motivo por el cual se ha decidido elegir este método es porque al principio de iniciar el proyecto es difícil recoger todos los requisitos que debe cumplir el proceso. Así siempre podremos volver atrás para corregir posibles errores de análisis o definición de los requisitos no capturados en estos documentos. Por lo que iremos desarrollando el proceso sobre la base que habremos creado con anterioridad, de esta manera nos obligamos a verificar y realizar un control de errores continuo en cada etapa y así podremos añadir más funcionalidades o corregir las ya creadas sin que ello repercuta críticamente en nuestro proyecto.

2.2. Planificación inicial

El proyecto constará de cinco grandes actividades, las cuales son las siguientes:

- **Análisis del producto:** Consistirá en analizar el sistema y el modo de trabajo además de la formación necesaria para crear la aplicación que hemos propuesto. La duración será de 130 horas.
- **Definición de requerimientos:** Base a la cual se construirá el software. Su objetivo es recoger y comprender el problema a abarcar, sus posibles soluciones, riesgos, beneficios, etc.; así como de describir y modelar abstractamente las funcionalidades que la aplicación tendrá. Su duración se estima en 40 horas.
- **Análisis y diseño:** En esta etapa, se realizará una visión lógica del sistema, refinando lo documentado en la definición de requerimientos. También se hará la visión física del mismo, es decir, el diseño de la base de datos, definición de tablas, de objetos de negocio, etc. La duración de esta etapa sumando las dos subtarefas de mantenimiento de balances y generación de los mismos será 100 horas subdivididas en dos fases, una para el primer formulario i la segunda etapa para el segundo formulario.
- **Construcción del software:** Aquí es donde se realizará la codificación de la aplicación. Esta etapa durará 210 horas. Primero se construirá el código del primer formulario y una vez codificado se pasará al 2º formulario.
- **Pruebas:** En esta actividad, se harán las pruebas definidas en el plan de pruebas, expuestos en el documento de análisis y diseño comprobado el correcto funcionamiento tanto del primer como del segundo formulario. Su duración total será de 40 horas.
- **Realización de la memoria:** Confección de la presente memoria. La duración será de 40 horas.

Teniendo en cuenta que inicialmente la duración del proyecto fuera de 560 horas, y dedicándole 4 diarias de lunes a viernes, la distribución de horas en las cinco grandes actividades es la siguiente:

	Nombre de tarea	Duración	Comienzo	Fin
1	Proyecto Contabilización mediante lector óptico	63,99 días	lun 09/11/09	lun 10/05/10
2	Inicio del proyecto: asignación y matrícula del proyecto	2 horas	lun 09/11/09	lun 09/11/09
3	Planificación	8,88 días	lun 09/11/09	jue 03/12/09
4	Estudio de viabilidad	40 horas	lun 09/11/09	lun 23/11/09
5	Aprobación Estudio Viabilidad (Punto de control)	1,03 horas	jue 03/12/09	jue 03/12/09
6	Análisis de la aplicación	20,5 días	lun 09/11/09	jue 24/12/09
7	Definición de requerimientos	39 horas	lun 23/11/09	mié 09/12/09
8	Aprobación definición de requerimientos (Punto de control)	1,03 horas	lun 21/12/09	lun 21/12/09
9	Análisis de datos (entorno karat)	20 horas	lun 09/11/09	lun 16/11/09
10	Análisis de funcionamiento (entorno eclipse-karat)	20 horas	lun 16/11/09	lun 23/11/09
11	Documentación del análisis	5 horas	mié 23/12/09	jue 24/12/09
12	Aprobación del análisis (Punto de control)	1 hora	jue 24/12/09	jue 24/12/09
13	Realización de un curso de formación en la aplicación karat y Java.	80 horas	jue 03/12/09	mar 22/12/09
14	Presentación inicial en la empresa	7,5 horas	lun 18/01/10	mar 19/01/10
15	Diseño de la aplicación	8,25 días	lun 11/01/10	mar 02/02/10
16	Diseño modular de la aplicación	20 horas	lun 11/01/10	vie 15/01/10
17	Diseño de la interfície y del borrador del formulario.	20 horas	lun 18/01/10	vie 22/01/10
18	Diseño de las pruebas (test)	20 horas	lun 25/01/10	vie 29/01/10
19	Documentación del diseño	5 horas	lun 01/02/10	mar 02/02/10
20	Aprobación del diseño (Punto de control)	1 hora	mar 02/02/10	mar 02/02/10
21	Desarrollo de la aplicación	23,13 días	mar 02/02/10	mié 07/04/10
22	Preparación entorno de desarrollo	20 horas	mar 02/02/10	mar 09/02/10
23	Módulo de adquisición de datos y funcionalidades de la aplicación	130 horas	mar 09/02/10	jue 25/03/10
24	Desarrollo de la interfície de facturas	35 horas	vie 26/03/10	mié 07/04/10
25	Test y pruebas	4,93 días	mié 07/04/10	mié 21/04/10
26	Pruebas unitarias	15 horas	mié 07/04/10	mar 13/04/10
27	Pruebas de integración a ekon.finanzas	15 horas	mar 13/04/10	lun 19/04/10
28	Pruebas d'estrés (incidèncias, riesgos)	5 horas	lun 19/04/10	mar 20/04/10
29	Documentación de desarrollo y test	3 horas	mar 20/04/10	mié 21/04/10
30	Aprobación del desarrollo y pruebas (Punto de control)	1,4 horas	mié 21/04/10	mié 21/04/10
31	Presentación final en la empresa	7,5 horas	mié 21/04/10	vie 23/04/10
32	Generación de documentos (memoria del proyecto)	40 horas	vie 23/04/10	vie 07/05/10
33	Cierre del proyecto	1 hora	vie 07/05/10	vie 07/05/10
34	Defensa del proyecto	5 horas	vie 07/05/10	lun 10/05/10

Figura 1: Planificación inicial proyecto

Teniendo como fecha de inicio el 9 de noviembre de 2009 y finalizando el 10 de mayo de 2010. A continuación vemos el diagrama de Gantt de la planificación:



Figura 2: Diagrama de Gant

2.3. Desviación respecto la planificación

Al hacer la planificación inicial, se tuvo en cuenta únicamente que siempre se trabajaría las mismas horas y que no surgiría ningún imprevisto ni complicación. A medida que avanzaba el proyecto había días en los que las horas dedicadas se incrementaban, y días en los que tenía que esperar la corrección del tutor.

Por otro lado, hubo problemas durante la fase de análisis de la aplicación y en el diseño de la aplicación, lo que hizo que se demorasen estas etapas. También en la fase de construcción del software, también hubo una desviación de tiempo, debido a las correcciones de código y estructuraciones del mismo.

En la siguiente tabla se muestra las desviaciones que sucedieron en las siguientes tareas:

Tarea	Tiempo estimado	Tiempo real	Desviación
Planificación	43	43	0
Análisis de la aplicación	173,5	182	+8,5
Diseño de la aplicación	66	72	+6
Desarrollo de la aplicación	185	241	+56
Test y Pruebas	39	35	-4
Memoria	53,5	30	-23,5
Total	560	603	43

Tal y como se puede apreciar ha habido una desviación en la duración total del proyecto en 43 horas, lo cual ha hecho que la duración total quedara fijada en 603 horas. Y la finalización del proyecto fuera el 21 de junio de 2010.

3. Definición de requerimientos

3.1. Sistema actual

3.1.1. Descripción

Actualmente *ekon finanzas* utiliza para el registro de facturas un proceso denominado *entrada de asientos*, basado en el uso de plantillas que dependerá del tipo de asiento que se deseará elaborar. A partir de la plantilla es posible introducir en una parrilla las diferentes entradas línea a línea y aplicarles, si se ha configurado para la línea a tratar, los datos de cartera, impuestos, bienes y analítica. Una vez se introducen y verifican los datos, es posible contabilizarlos, haciendo efectivo el borrador del asiento que se ha creado.

De esta forma, permite realizar apuntes contables indicando cómo y dónde se aplicará cada una de las partes en que se desglosará la entrada de una factura, registrando éstas en sus correspondientes cuentas, las cuales están asociadas a planes contables para facilitar su agrupación y consulta.

3.1.2. Diagnóstico

El producto *ekon finanzas* es un producto muy elaborado en cuanto al registro de facturas, por las prestaciones de que dispone y el nivel de detalle de éstas. Sin embargo, para poder tratarlo y crear nuevos asientos es necesario conocer detalladamente el funcionamiento del producto y saber qué plantilla y tratamiento concreto usar en cada tipo de entrada que se desee realizar. Además, tiene la limitación de que no hay ningún proceso que permita entrar datos a través de un lector óptico.

3.2. Sistema propuesto

3.2.1. Objetivos

El objetivo del proyecto es crear un nuevo proceso para el registro de facturas que permita entrar facturas tradicionales en papel sin necesidad de teclear nada, sólo pasando el lector óptico por encima de unos campos concretos de la factura tradicional de papel. Esto nos generará un borrador, gracias al cual se podrá crear asientos con facilidad sin requerir conocimientos avanzados de *ekon finanzas*.

3.2.2. Objetos

El proceso constará de dos formularios: *pantalla previa* y *entrada automática*.

- El formulario pantalla previa estará estructurado en 3 secciones: *Datos comunes a todas las facturas, restricciones y valores por defecto*.
- El formulario *entrada automática* estará estructurado en cinco secciones: *lectura de importes, entrada de datos, datos factura, desgloses y asientos generados*.

3.2.2.1. Pantalla previa

Inicialmente se abrirá un formulario con datos básicos para la entrada automática.

The screenshot shows the 'Entrada automática' window. It includes the following fields and controls:

- Entidad:** A text input field.
- Libro:** A text input field.
- Asignar la fecha de factura como fecha contable:** A checkbox.
- Fecha contable:** A date input field.
- Plantilla:** A text input field with a folder icon on the right.
- Carpeta destino:** A text input field with a folder icon on the right.
- Restricciones:** A section containing:
 - Ejercicio:** A text input field.
 - Periodo:** A text input field.
- Valores por defecto:** A section containing:
 - Diario:** A text input field.
 - Tipo doc.:** A text input field.
- Contabilización directa:** A checkbox.
- Buttons:** 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons.

Figura 3: Formulario Pantalla Previa

Datos comunes a todas las facturas

This image shows the top portion of the 'Entrada automática' window, highlighting the common data fields for all invoices:

- Entidad:** Text input field.
- Libro:** Text input field.
- Asignar la fecha de factura como fecha contable:** Checkbox.
- Fecha contable:** Date input field.
- Plantilla:** Text input field with a folder icon.
- Carpeta destino:** Text input field with a folder icon.

Figura 4: Pantalla Previa: Datos comunes a todas las facturas

- **Entidad:** Entidad con la cual trabajaremos.
- **Libro:** Libro del proveedor/cliente de la entidad seleccionada.
- **Asignar la fecha de factura como fecha contable:** Check que asigna la fecha de la factura como fecha usada para los efectos contables.
- **Fecha contable:** Si no se ha marcado el check anterior, se entrará la fecha contable para todas las facturas.
- **Plantilla:** Plantilla que se desea utilizar al introducir la factura.
- **Carpeta de destino:** Carpeta de destino del borrador generado.

Restricciones

Figura 5: Pantalla Previa: Restricciones

- **Ejercicio:** Si se especifica, solo permite facturas de ese ejercicio.
- **Periodo:** Si se especifica, solo permite facturas de ese periodo.

Valores por defecto

Figura 6: Pantalla Previa: Valores por defecto

- **Diario:** Código del diario usado en la factura. En caso que la plantilla no tenga uno asignado cogerá éste.
- **Tipo doc.:** Corresponderá al tipo de documento que crearemos. En caso que la plantilla no tenga uno asignado cogerá éste.
- **Contabilización directa:** Check que indica si se quiere contabilizar la factura entrada directamente al terminar de leerla o no. Por defecto estará siempre marcada.

3.2.2.2. Entrada automática

Los datos leídos con el lector y que rellenarán el formulario (figura 7) son los que aparecerán en el asiento que estamos generando.

En la sección de asientos generados aparecerán los asientos para poder visualizarlos o modificarlos.

En la pantalla podemos distinguir varias zonas:

- Lectura de importes.
- Entrada de datos.
- Datos factura.
- Desgloses.
- Asientos generados.

[CCS] Entrada automática

ekon

Lectura de importes

Primero

Segundo

Tercero

Cuarto

Entrada de datos

Nombre Control

Multi-IVA ☐

Fin multi-IVA

Contabilizar

Datos factura

Nº factura

Fecha factura

Fecha contable

Divisa

Cuenta

Descripción

CIF/NIF

Total factura

Lectura de importes (IVA) IRPF Vencimientos Contrapartida (Gastos/Ingresos)

Cód. IVA	Mod.	Tp. impt.	Base	%IVA	Cuota	%Rec.	Cuota Rec.	Total	Cuenta
----------	------	-----------	------	------	-------	-------	------------	-------	--------

Asientos generados

Diario	N. Asiento	Tp. doc.	F. Contable	F. Factura	Referencia	Núm. doc.	Total
--------	------------	----------	-------------	------------	------------	-----------	-------

Ver asiento

Figura 7 Formulario Entrada Automática

Lectura de importes

Lectura de importes

Primero

Segundo

Tercero

Cuarto

Figura 8: Entrada Automática: Lectura de importes

Este marco nos indicará el orden de lectura para los importes. Las opciones posibles serán: Base, IVA, Recargo, Total.

No es necesario informar los 4 órdenes, con uno es suficiente.

Si sólo se informa un orden, el resto de importes se calcularán a partir del código de IVA informado en la definición de la Cuenta Contable (Cliente/Proveedor).

Entrada de datos

Entrada de datos

Nombre Control

Multi-IVA

☐

Fin multi-IVA

Contabilizar

Figura 9: Entrada Automática: Entrada de datos

Habr  un control principal donde se visualizar  el dato actual le do por el lector  ptico. Como “Nombre Control” se ir  visualizando el dato que se entra en cada momento, es decir, si es “N  Factura”, “Fecha factura”, “Cliente/Proveedor”, “Importe...”...

Tambi n se podr  indicar si la factura tiene Multi-IVA y cuando se acaban de entrar los diferentes IVAs.

Y tendr  el bot n para Contabilizar el asiento entrado. La contabilizaci n siempre consta de la Verificaci n previa. Si se produjera cualquier incidencia, se visualizar .

Datos factura

Datos factura

N  factura

Cuenta

Fecha factura

Descripci n

Fecha contable

CIF/NIF

Divisa

Total factura

Figura 10: Entrada Autom tica: Datos factura

En este marco se ir n informando los datos referentes a la factura y cliente/proveedor a partir de la informaci n entrada en el marco de Entrada de datos.

Desgloses

En todas las parrillas de desglose se podr n informar los datos manualmente.

- Desglose de IVA

IVA										
IRPF										
Vencimientos										
Contrapartida (Gastos/Ingresos)										
C�d. IVA	Mod.	Tp. impt.	Base	%IVA	Cuota	%Rec.	Cuota Rec.	Total	Cuenta	

Figura 11: Entrada Autom tica: Desglose de IVA

A partir de la información entrada y los datos de la cuenta contable (cliente/proveedor), se irán generando las líneas de IVA.

- **Desglose de IRPF**

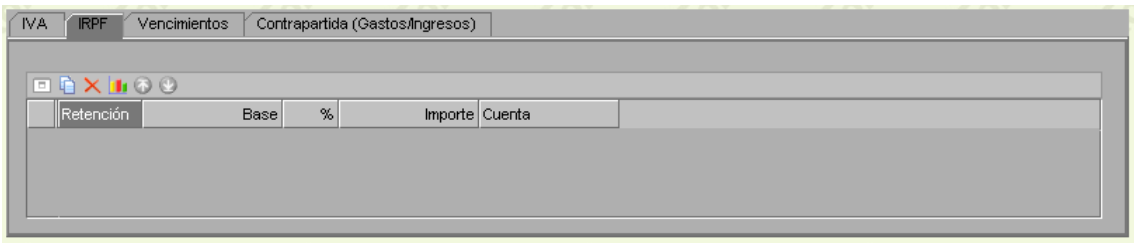


Figura 12: Entrada Automática: Desglose de IRPF

En la pestaña de IRPF, si el declarado asociado a la cuenta contable (cliente/proveedor) tiene retenciones, se calcularán automáticamente.

- **Desglose de Vencimientos**

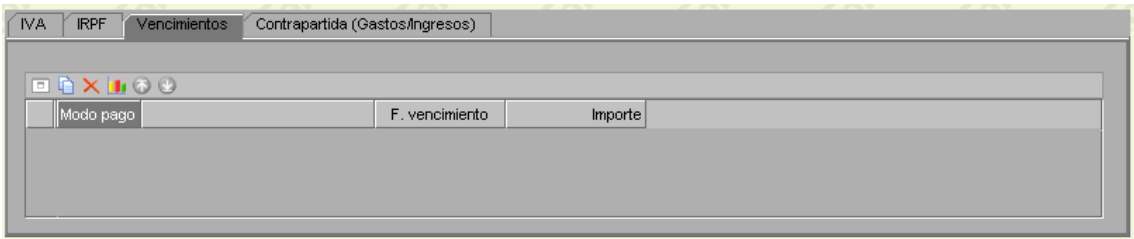


Figura 13: Entrada Automática: Desglose de Vencimientos

A partir de la cuenta contable (cliente/proveedor) y sus localizaciones se calcularán los vencimientos, siempre y cuando la cuenta tenga tratamiento de cartera.

- **Desglose de Contrapartida (Gastos/Ingresos)**

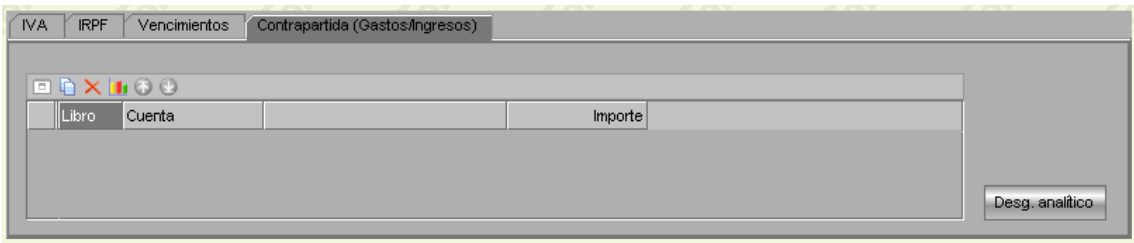


Figura 14: Entrada Automática: Desglose de Contrapartida

A partir de la cuenta contable (cliente/proveedor) se calcularán las contrapartidas.

Si la cuenta de contrapartida tiene marcada la opción de Tratamiento analítico, se activa el botón ‘Desglose analítico’ y se calcula el desglose analítico. Pulsando el botón ‘Desglose analítico’ se accede a una ventana donde se visualiza éste (figura 15).

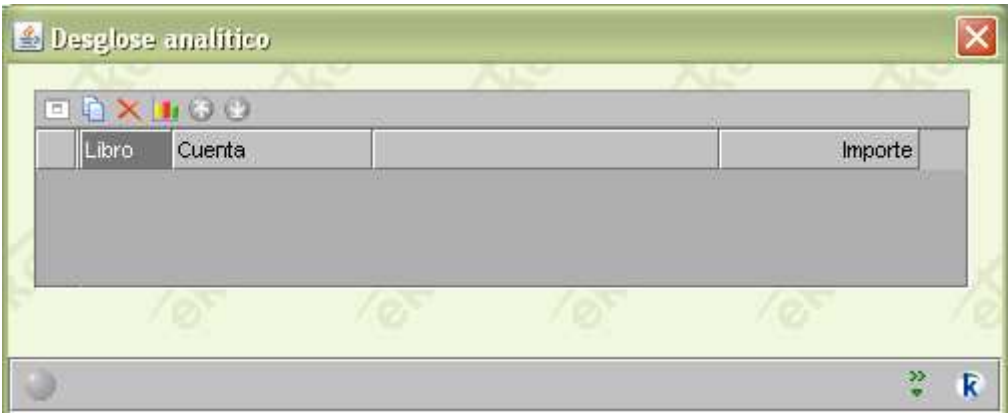


Figura 15: Entrada Automática: Desglose analítico

Asientos generados

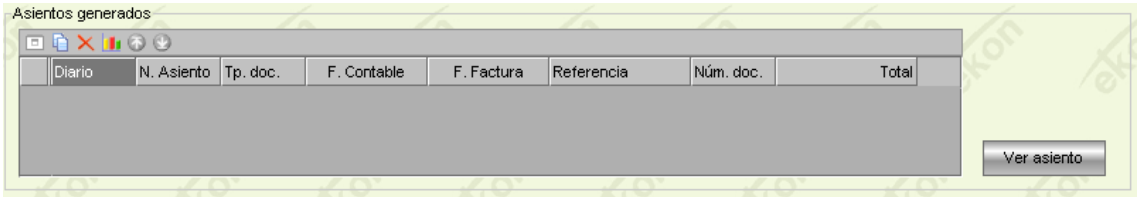


Figura 16: Entrada Automática: Asientos generados

En este marco, se visualizan los asientos que se han ido generando durante la sesión.

Al pulsar el botón ‘Ver asiento’, permite visualizar el asiento seleccionado en la parrilla. Desde la visualización también se permitirá modificar si se desea.

3.2.3. Reglas de gestión

Regla de gestión	
1	En la lectura de importes no es necesario informar los 4 órdenes, con uno es suficiente.
2	La contabilización siempre consta de la Verificación previa, si se produjera cualquier incidencia, se visualizaría.
3	En caso de que un campo no se haya podido introducir se podrá introducir manualmente.
4	Si una lectura con el lector resulta errónea se tiene que poder entrar los datos manualmente y una vez entrados poder realizar la siguiente entrada con el lector o manualmente.
5	Habrà un control principal donde se visualizarà el dato actual leído por el lector óptico. Como “Nombre Control” se irá visualizando el dato que se entra en cada momento, es decir, si es “Nº Factura”, “Fecha factura”, “Cliente/Proveedor”, “Importe...”...
6	En todas las parrillas de desglose se podrán informar los datos manualmente.
7	A partir de la información entrada y los datos de la cuenta contable (cliente/proveedor), se irán generando las líneas de IVA.
8	En la pestaña de IRPF, si el declarado asociado a la cuenta contable (cliente/proveedor) tiene retenciones se calcularán automáticamente.

	Regla de gestión
9	A partir de la cuenta contable (cliente/proveedor) y sus localizaciones, se calcularán los vencimientos siempre y cuando la cuenta tenga tratamiento de cartera.
10	A partir de la cuenta contable (cliente/proveedor) se calcularán las contrapartidas.
11	Desde la visualización del asiento también se permitirá modificar.
12	Para validar la fecha se tendrá que comprobar que cumpla un formato valido y aceptado.
13	Para validar la sección del proveedor/cliente se tendrá que comprobar que exista en la base de datos y se hayan completado correctamente los campos del proveedor/cliente.

3.2.4. Funcionalidades

Las funcionalidades que ofrecerá serán las siguientes:

- Los formularios
 - Tipos de formularios
 - Pantalla previa
 - Fecha contable
 - Restricciones
 - Valores por defecto
 - Formulario principal
 - Lectura de importes
 - Entrada de datos
 - Datos factura
 - Asientos generados

3.2.5. Otros requerimientos no funcionales

- Restricciones de diseño

Al tratarse de un nuevo proceso de facturación para *ekon finanzas* debe funcionar acorde con el producto, utilizando las mismas estructuras de datos y terminología contable. De la misma forma que deberá almacenar los datos facturados siguiendo el mismo patrón que la entrada de asientos original del producto.

El proceso debe poder verificarse con la utilidad de que dispone el producto antes de contabilizar los datos introducidos. El proceso tiene que ser desarrollado en Java y tiene que tener un alto grado de calidad y fiabilidad.

- Objetivos de diseño

El objetivo del proyecto es realizar un proceso de entrada de facturas amigable para el usuario, por lo tanto, el formulario que se mostrará en el proceso debe ser claro y entendedor a primera vista:

- El usuario debe poder interpretar cada uno de los campos que debe introducir y entender la funcionalidad del programa sin realizar un proceso de abstracción complejo.

- Los datos introducidos deben poder modificarse, de manera que cuando se esté usando el proceso se puedan realizar rectificaciones sobre la marcha en caso que hubiera alguna lectura errónea por parte del lector óptico.

3.2.6. Requerimientos de Software

- Requerimientos de rendimiento

El proceso debe cumplir con los requerimientos del producto *ekon*, es decir, debe funcionar tanto en cliente/servidor como en web mediante Karat.

- Arquitectura de software y de hardware

Tendrá los mismos requerimientos que los existentes en el producto *ekon finanzas*, ya que el proceso que se va a desarrollar es un módulo para el producto. Los cuales consisten en normas a la hora de codificar la aplicación y tablas concretas donde extraer la información de finanzas que se desee.

- Condicionantes de implantación y funcionamiento

La funcionalidad está contemplada en la versión 8.0 del producto *karat*.

3.2.7. Plan de validación

- Supervisión, corrección y validación del documento de definición de requerimientos por el tutor.
- Verificación de la estructuración del modelo de datos del sistema.
- Supervisión, corrección y validación del análisis y del diseño por el tutor.
- Seguimiento de la construcción del proceso por el validador técnico.
- Realización de pruebas por parte del programador
- Corrección de los errores por el programador y verificación de la construcción por parte del validador técnico.
- Realización de pruebas de integración y corrección de errores.
- Realización de pruebas de sistema y corrección de errores.
- Validación del proceso por parte del tutor.

3.3. Costes, beneficios y riesgos

Los beneficios que se obtendrán con el proyecto son la obtención de un proceso amigable para el usuario, que una vez integrado en el producto le permitirá introducir facturas usando el lector óptico a partir de leer unos pocos datos de la factura en papel con el lector, permita crear un borrador, corregir el borrador en caso de modificaciones y contabilizar la factura.

Los costes del programa, serán la dedicación por mi parte hasta que finalice el proyecto.

El riesgo más importante es el poco conocimiento de karat, lo que puede hacer incrementar el número de horas de programación del software.

4. Análisis y diseño

4.1. Introducción

El proceso a desarrollar consistirá en una utilidad para el registro de facturas que permita entrar facturas tradicionales en papel sin necesidad de teclear nada, sólo pasando el lector óptico por encima de unos campos concretos de la factura tradicional en papel. Esto nos generará un borrador, gracias al cual se podrá crear asientos con facilidad sin requerir *conocimientos* avanzados de *ekon finanzas*.

Una vez se hayan introducido los datos que se desean computar se realizará una conversión, con la finalidad de pasar la información introducida a un formato estándar utilizado por **ekon finanzas**.

El objetivo de nuestro proyecto es que el producto final generado por nuestro proceso sea una factura contabilizada o un borrador.

4.2. Contabilización mediante lector óptico

4.2.1. Introducción de datos en Pantalla previa

Figura 17: Pantalla Previa

Situada en la parte superior de la cabecera encontraremos los siguientes campos:

- **Entidad:** Entidad con la cual trabajaremos.
- **Libro:** Libro del proveedor/cliente de la entidad seleccionada.
- **Asignar la fecha de factura como fecha contable:** Check que asigna la fecha de la factura como fecha usada para los efectos contables.
- **Fecha contable:** Si no se ha marcado el check anterior, se entrará la fecha contable para todas las facturas.
- **Plantilla:** Plantilla que se desea utilizar al introducir la factura.
- **Carpeta de destino:** Carpeta de destino del borrador generado.

También dispondremos de varias secciones que nos permitirán configurar la entrada automática de datos de la factura:

- **Restricciones**

El **campo ejercicio**, si se especifica, sólo permite facturas de ese ejercicio. Y el **campo periodo**, si se especifica, sólo permite facturas de ese periodo.

- **Valores por defecto**

Podremos introducir datos referentes a la identificación de la factura que vamos a crear. Tales como el código del **diario** usado en la factura, (en caso que la plantilla no tenga uno asignado cogerá éste) y el **tipo de documento** que crearemos, (en caso que la plantilla no tenga uno asignado cogerá éste). También habrá un check, llamado **contabilización directa**, que indicará si se quiere contabilizar la factura entrada directamente al terminar de leerla o no. Por defecto estará siempre seleccionado.

4.2.2. Introducción de datos en la entrada automática

Dispondremos de varias secciones (figura 18) que nos permitirán configurar la factura:

- **Lectura de importes**

Este marco nos indicará el orden de lectura para los importes. Las opciones posibles serán: Base, IVA, Recargo, Total.

No es necesario informar los 4, con uno es suficiente. Si sólo se informa un orden, el resto de importes se calcularán a partir del código de IVA informado en la definición de la Cuenta Contable (Cliente/Proveedor).

- **Entrada de datos**

Habrà un control principal donde se visualizarà el dato actual leído por el lector óptico. En “Nombre Control” se irá visualizando el dato que se debe entrar en cada momento, es decir, si es “Nº Factura”, “Fecha factura”, “Cliente/Proveedor”, “Importe...”...

También se podrá indicar si la factura tiene Multi-IVA y cuando se acaban de entrar los diferentes IVAs.

Figura 18 Entrada Automática

También tendrá el botón para Contabilizar el asiento entrado. La contabilización siempre consta de la Verificación previa. Si se produjera cualquier incidencia, se visualizaría información al respecto de ella.

- **Datos factura**

En este marco se irán informando los datos referentes a la factura y cliente/proveedor a partir de la información entrada en el marco de Entrada de datos. Debemos especificar el **número de factura**, la **fecha de la factura**, la **fecha contable**, la **divisa**, la **cuenta**, la **descripción**, el **CIF/NIF** y el **total factura**.

Siempre mostrará la **divisa** con la que trabajaremos. Por defecto se utilizará la divisa base identificada en la entidad, sin embargo es posible utilizar otra de las divisas permitidas mediante el botón cambio de **divisa**, identificado con el siguiente icono.

Figura 19: Entrada Automática: Divisa

4.2.2.1. Cálculo de IVA

La primera línea de IVA se generará automáticamente si existe **código de IVA** asociado al proveedor/cliente para el que se está entrando la factura (en la definición de la cuenta contable).

El campo **Cód. IVA** corresponderá al código de IVA identificado en la cuenta de proveedor/cliente. Este código de IVA, juntamente con el **plan contable general** de la entidad, nos determinará la **modalidad, tipo de impuesto, cuenta general y cuenta de recargo** de la línea. A partir del **tipo de impuesto** y la **fecha factura** se generará el **%IVA** y **%Rec.** La **base imponible** de esta primera línea será calculada usando el **total factura** introducido en la sección **datos factura**. A partir de esta base imponible y los **porcentajes de IVA** y de **recargo** podremos obtener, la **cuota general** y **cuota de recargo**. La **cuota de recargo** únicamente se calculará si existe **% de recargo** y la normativa de la entidad lo permite. A partir de todos estos datos se calculará el **total** por cada línea de IVA.

En caso de insertar una nueva línea, el valor del campo **Cód. IVA** será el correspondiente al código de IVA secundario, también en la cuenta de proveedor/cliente. El resto de campos se calcularán de la misma forma que la primera línea, con un importe igual a la base imponible obtenida.

La mayoría de estos campos pueden modificarse manualmente, con excepción de los que son de carácter informativo, provocando los correspondientes recálculos del valor de los campos de la línea que dependan del campo modificado.

4.2.2.2. Cálculo del IRPF

La primera línea de IRPF se calculará automáticamente si el proveedor/cliente escogido tiene retenciones en la cuenta contable asociada.

El campo **retención** vendrá determinado por el código IRPF asociado al declarado, el **%** se obtendrá del campo **retención** y la **fecha factura**. La base imponible de esta primera línea corresponderá al de la primera línea de IVA, ya que esta **base imponible** se calcula a partir del **importe total**, y del **%** del IRPF. Mediante la **base imponible** y el **%** se obtendrá el **importe** de retención.

La **base imponible** de las posteriores líneas insertadas será, por defecto, la diferencia entre la base imponible total y la suma de la base imponible de las líneas de IRPF.

La **cuenta** de la línea de IRPF se informará por defecto o el usuario podrá escogerla manualmente.

4.2.2.3. Cálculo de vencimientos

Habrà una línea por cada vencimiento y estarán formadas por los siguientes campos:

- **Método de pago:** El método que se utilizará para pagar este vencimiento concreto.
- **F.vencimiento:** La fecha en que se hará efectivo el vencimiento.
- **Importe:** El importe asociado a este vencimiento.

Los vencimientos se calcularán automáticamente a partir de la cuenta contable y sus localizaciones siempre y cuando la cuenta tenga tratamiento de cartera, sin embargo el usuario los podrá modificar como desee manualmente.

4.2.2.4. Cálculo de contrapartida

A partir de la cuenta contable se calcularán las contrapartidas. Para acceder al desglose analítico de una de las líneas de la factura, deberemos situarnos sobre la pestaña **Contrapartida (Gastos/ingresos)**, en una línea de la parrilla y pulsar el botón **Desg.analítico** situado en la parte inferior derecha de la ventana.

Únicamente se mostrarán líneas analíticas si la **cuenta** de la línea con la que deseamos trabajar dispone de **imputación analítica**. De lo contrario, un mensaje informará de ello.

En caso de que la línea disponga de imputación analítica definida en su cuenta, se mostrarán una o más líneas analíticas, dependiendo de si la cuenta tiene asociado un **desglose analítico**. Si la cuenta de contrapartida tiene marcada la opción de Tratamiento analítico, se activará el botón Desglose analítico y se mostrarán tantas líneas como hayan definidas en el desglose de la cuenta, junto con su correspondiente **libro analítico**, **cuenta** y la fracción de **importe en divisa** correspondiente al **% de reparto** identificado.

Si la cuenta dispone de **imputación analítica** pero no de **desglose analítico** se mostrará una única línea con el **libro analítico**, la **cuenta** y un **importe en divisa** igual al importe de la línea de la factura.

En cada línea dispondremos de un campo de **descripción** donde podremos especificar la finalidad u origen de cada una de las líneas analíticas.

Cuando se modifique la cuenta o el importe en una línea de factura, se recalculará la analítica automáticamente.

4.2.2.5. Asientos generados

En este marco, se visualizan los asientos que se han ido generando durante la sesión.

Habrà una línea por cada asiento generado y estarán formadas por los siguientes campos:

- **Diario:** Código identificador del diario de la factura
- **NºAsiento:** Número de asiento generado
- **Tp.doc:** Código identificador del tipo de documento
- **F.Contable:** Fecha contable de la factura
- **F.Factura:** Fecha de la factura
- **Referencia:** Referencia del documento
- **Núm.doc.:** Identificador del documento
- **Total:** Importe total del asiento generado

Los asientos se generarán automáticamente a partir de los datos entrados con el lector. Al seleccionar una línea y pulsar el botón **Ver asiento**, permite visualizar el asiento seleccionado en la parrilla. Desde la visualización también se permitirá modificar, si se desea.

4.3. Modelo físico

4.3.1. Nueva tabla

Diagrama de las diferentes tablas con los campos necesarios que debemos crear para conseguir el correcto funcionamiento del proceso a desarrollar.

pf_cg_fra_usu
xusuario_id
xentidad_id
xlibro_id
xfacfra_asig
xfecha_contable
xplantilla_id
xplant_dest
xejercicio_id
xperiodo
xdiario_id
xtipo_doc
xcontab_direc
xlectimp_1
xlectimp_2
xlectimp_3
xlectimp_4

Figura 20: Tabla bases de datos

4.3.2. Descripción de componentes

4.3.2.1. Tablas

pf_cg_fra_usu

Descripcion: tabla que guardará los datos entrados en el formulario de datos previos:

Campos

Nombre	Descripción	Tipo
xusuario_id	Código identificador del usuario	TEXT
xentidad_id	Código identificador de la entidad contable	TEXT
xlibro_id	Código identificador del libro contable	TEXT
xfacfra_asig	Asignar la fecha de factura como fecha contable	INTEGER
xfecha_contable	Fecha contable de la factura	DATE
xplantilla_id	Código de la plantilla que se utilizará para generar el asiento	LONG
xplant_dest	Carpeta de destino del borrador generado	LONG
xejercicio_id	Ejercicio contable	TEXT
xperiodo	Periodo contable	INTEGER
xdiario_id	Código identificador del diario de la factura	TEXT
xtipo_doc	Código del tipo de documento a tratar	TEXT
xcontab_direc	Indica si los documentos generados se verifican y contabilizan directamente	INTEGER
xlectimp_1	1º importe a leer con el lector óptico	INTEGER
xlectimp_2	2º importe a leer con el lector óptico	INTEGER
xlectimp_3	3º importe a leer con el lector óptico	INTEGER

xlectimp_4	4° importe a leer con el lector óptico	INTEGER
------------	--	---------

4.3.2.2. Consultas base (pf_cg_fra_usu)

Se proyectan los campos de la tabla pf_cg_fra_usu.

```
SELECT xusuario_id, xentidad_id, xlibro_id, xfecha_asig, xfecha_contable, xplantilla_id,
xplant_dest, xejercicio_id, xperiodo, xdiario_id, xtipo_doc, xcontab_direc, xlectimp_1,
xlectimp_2, xlectimp_3, xlectimp_4 FROM pf_cg_fra_usu ORDER BY xusuario_id,
xentidad_id
```

4.3.2.3. Listas de sistema (syf_lectura_imp)

Código	Descripción
1	Base
2	IVA
3	Recargo
4	Total

4.3.2.4. Parámetros pf_ent_activa (Tratamiento Entidad activa)

Esta entrada de parámetros nos permitirá que cuando la entidad activa sea cambiada, nuestro formulario muestre la entidad actual con la que estemos trabajando.

Proceso	Control 1	Campo	Tp.
pf_cg_fra_usu	txtentidad_id	xentidad_id	2

cpt_tablasentidad (Tabla a tratar al borrar una entidad)

tetabla	teentite.
pf_cg_fra_usu	xentidad_id

4.4. Especificaciones de procesos

4.4.1. Contabilización mediante lector óptico

4.4.2. Datos iniciales

El objeto de negocio pf_cg_fra_ini será el esqueleto de la pantalla previa, a él está vinculado el formulario que usaremos de interfaz gráfica en la pantalla previa y el proyecto en Java.

Desde el formulario introduciremos datos en los campos y veremos el funcionamiento del proceso, mediante las funciones que habremos codificado en el proyecto Java.

El objeto de negocio dispone de tres secciones que utilizaremos en nuestro proceso:

Objeto de negocio <pf_cg_fra_ini>

4.4.2.1. Paneles

Los paneles son las secciones en que dividiremos los conjuntos de campos que serán definidos en el objeto de negocio. Cada panel está relacionado con la función que tienen los campos que contiene. En nuestro caso, cada panel corresponde a una de las tablas que hemos creado, por ello dispondrán de una finalidad específica.

El objeto de negocio pf_cg_fra_ini estará formado por un panel, al que se vincularán las tablas anteriormente definidas. En él se crearán todos los campos que se mostrarán en el formulario y otros que no se mostrarán pero serán necesarios para almacenar valores, realizar cálculos y operaciones necesarias para el buen funcionamiento del proceso.

- Pantalla previa <HEADER>

En todo objeto de negocio hay un <HEADER>, que es el panel principal del objeto de negocio. Identificará a la cabecera de nuestro proceso y a la mayor parte de los campos del formulario base. Estará referenciado por el resto de paneles.

Controles

Los controles necesarios para este panel son los siguientes:

Control	Descripción
ID Usuario	Código identificador del usuario
Entidad	Código identificador de la entidad contable
Libro	Código identificador del libro de proveedor/cliente
Asignar la fecha de factura como fecha contable	Asignar la fecha de factura como fecha contable
Fecha contable de la factura	Fecha contable de la factura
Plantilla	Código de la plantilla que se utilizará para generar el asiento
Carpeta destino	Carpeta de destino del borrador generado
Ejercicio	Ejercicio contable al que corresponde la factura
Periodo	Periodo del ejercicio contable al que corresponde la factura
Diario	Código identificador del diario de la factura
Tipo documento	Código del tipo de documento a tratar
Contabilización directa	Indica si los documentos generados se verifican y contabilizan directamente

4.4.2.2. Consultas dependientes

Las consultas dependientes son consultas internas del objeto de negocio que utiliza para acceder a determinados datos almacenados en tablas del repositorio. Para poder crear una consulta dependiente, es necesario antes disponer de una consulta base a la tabla que deseamos consultar.

Una vez sabemos la consulta base que deseamos utilizar, especificaremos los parámetros que condicionarán la búsqueda, relacionando el campo de la tabla en la que buscaremos con una constante, variable de entorno o un campo de los que tenemos en uno de los paneles de nuestro objeto de negocio. De esta manera, la consulta filtrará los resultados según el valor que le hemos introducido.

Las consultas dependientes son especialmente usadas como MDQO's (consultas en una tabla de la base de datos partiendo del valor de un control) en LOOK UP's (menu de selección del valor de un campo viendo los valores de la base de datos que cumplen las restricciones que le imponemos) o campos calculados, asociados a campos del objeto de negocio. Por lo general tendremos dos consultas base para cada LookUp, que mostrarán los mismos datos pero ordenados por código o por descripción.

Nuestro objeto de negocio (pf_cg_fra_ini) usará las siguientes consultas dependientes:

mdqo_entidad_c

Consulta:	cpt_he_gen_vdl		
Dependencias:	Campo	Tipo	Valor
	heentite	Control	txtentidad_id

mdqo_entidad_d

Consulta:	cpt_he_gen_vdl_d		
Dependencias:	Campo	Tipo	Valor
	heentite	Control	txtentidad_id

mdqo_libro_c

Consulta:	cpt_hn_gty_vdl		
Dependencias:	Campo	Tipo	Valor
	hnentite	Control	txtentidad_id
	hntypent	Control	txtlibro_id

mdqo_libro_d

Consulta:	cpt_hn_gty_vdl_d		
Dependencias:	Campo	Tipo	Valor
	hnentite	Control	txtentidad_id
	hntypent	Control	txtlibro_id

mdqo_ejercicio_c

Consulta:	cpt_ca_cal		
Dependencias:	Campo	Tipo	Valor
	cacal	Control	txtejercicio_id

mdqo_ejercicio_d

Consulta:	cpt_ca_cal_d		
Dependencias:	Campo	Tipo	Valor
	cacal	Control	txtejercicio_id

mdqo_periodo_c

Consulta:	cpt_pe_cal		
Dependencias:	Campo	Tipo	Valor
	pecal	Control	txtejercicio_id
	penumper	Control	txtperiodo

mdqo_periodo_d

Consulta:	cpt_pe_cal_d		
Dependencias:	Campo	Tipo	Valor
	pecal	Control	txtejercicio_id
	penumper	Control	txtperiodo

mdqo_diario_c

Consulta:	pf_diarios_ent_vdl		
Dependencias:	Campo	Tipo	Valor
	xentidad_id	Control	txtentidad_id
	xcjcodjou	Control	txtdiario_id

mdqo_diario_d

Consulta:	pf_dia_ent_vdl_d		
Dependencias:	Campo	Tipo	Valor
	xentidad_id	Control	txtentidad_id
	xcjcodjou	Control	txtdiario_id

mdqo_tipo_doc_c

Consulta:	cpt_te_vdl		
Dependencias:	Campo	Tipo	Valor
	tecode	Control	txttipo_doc

4.4.2.3. Formulario Datos iniciales < pf_cg_fra_ini >

La siguiente captura corresponde al formulario de **Datos iniciales**:

Figura 21: Pantalla Previa

4.4.3. Entrada Factura (Lector óptico)

El objeto de negocio `pf_cg_factura` será el esqueleto del proceso, a él está vinculado el formulario que usaremos de interfaz gráfica en la entrada automática.

Desde el formulario introduciremos datos en los campos y veremos el funcionamiento del proceso, mediante las funciones que habremos codificado en el proyecto Java.

El objeto de negocio dispone de cinco secciones que utilizaremos en nuestro proceso:

Objeto de negocio <pf_cg_factura> 4.4.3.1. Paneles

Los paneles son las secciones en que dividiremos los conjuntos de campos que serán definidos en el objeto de negocio. Cada panel está relacionado con la función que tienen los campos que contiene. En nuestro caso, cada panel corresponde a una de las tablas que hemos creado, por ello dispondrán de una finalidad específica.

El objeto de negocio `pf_cg_factura` estará formado por seis paneles, a los que se vincularán las tablas anteriormente definidas. En ellos se crearán todos los campos que se mostrarán en el formulario y otros que no se mostrarán pero serán necesarios para almacenar valores, realizar cálculos y operaciones necesarias para el buen funcionamiento del proceso.

- **Entrada automática (Lector óptico) <HEADER>**

En todo objeto de negocio hay un <HEADER>, es el panel principal del objeto de negocio. Identificará a la cabecera de nuestro proceso y la mayor parte de los campos del formulario base. Estará referenciado por el resto de paneles.

Controles

Los controles necesarios son los siguientes:

Control	Descripción
ID Usuario	Código identificador del usuario
Primero	1r importe a leer con el lector óptico
Segundo	2º importe a leer con el lector óptico
Tercero	3r importe a leer con el lector óptico
Cuarto	4º importe a leer con el lector óptico
Nombre Control	Nombre del control que vamos a leer con el lector óptico
Multi-IVA	Seleccionar Multi-IVA en la factura
Número de factura	Número de factura
Fecha de la factura	Fecha de la factura
Fecha contable de la factura	Fecha contable
Código divisa	Tipo divisa
Código de la cuenta bancaria	Código identificador de la cuenta de proveedor/cliente
Descripción	Descripción de la factura
CIF/NIF	Código identificador del proveedor/cliente
Total factura	Importe total de la factura

- **pnl_iva**

Descripción: Líneas de IVA de la factura

Controles

Los controles utilizados son los siguientes:

Control	Descripción
Línea de IVA	Número de línea de IVA según tipo de IVA
Código de IVA	Código de IVA
Modalidad de IVA	Código identificativo de modalidad de IVA
Tipo impositivo	Tipo impositivo de la línea de IVA
Base imponible de la línea de IVA	Base imponible de la línea de IVA
Porcentaje de IVA	Porcentaje general de la línea de IVA
Cuota	Cuota de la línea de IVA
Porcentaje de recargo	Porcentaje de recargo de la línea de IVA
Cuota de recargo	Cuota de recargo de la línea de IVA
Total línea IVA	Importe total de la línea de IVA
Código de la cuenta bancaria	Código identificador de la cuenta de proveedor/cliente

- pnl_irpf

Descripción: Líneas de IRPF de la factura

Controles

Los controles utilizados son los siguientes:

Control	Descripción
Línea de IRPF	Número de línea de IRPF según tipo de IRPF
Retención del IRPF	Código de retención de IRPF
Base imponible de la línea de IRPF	Base imponible de la línea de IRPF
Porcentaje del IRPF	Porcentaje general de la línea de IRPF
Importe IRPF	Importe de la línea de IRPF
Código de la cuenta bancaria	Código identificador de la cuenta de proveedor/cliente

- pnl_vencimientos

Descripción: Líneas de vencimientos de la factura

Controles

Los controles utilizados son los siguientes:

Control	Descripción
Línea de vencimientos	Número de línea de IVA según tipo de IVA
Modo de pago	Modo de pago
Fecha de vencimiento	Fecha de vencimiento
Importe Vencimientos	Importe de la línea de vencimientos

- pnl_contrapartida

Descripción: Líneas de contrapartida de la factura

Controles

Los controles utilizados son los siguientes:

Control	Descripción
Libro	Código identificador del libro de proveedor/cliente
Código de la cuenta bancaria	Código identificador de la cuenta de proveedor/cliente
Importe Contrapartida	Importe de la línea de contrapartida

- pnl_asientos

Descripción: Líneas de asientos de la factura

Controles

Los controles utilizados son los siguientes:

Control	Descripción
Diario	Código identificador del diario de la factura
Número asiento	Número de línea de asiento
Tipo documento	Código del tipo de documento a tratar
Fecha contable de la factura	Fecha contable de la factura
Fecha de la factura	Fecha de la factura
Referencia	Referencia del documento
Número documento	Código identificador del documento
Total línea asiento	Importe total de la línea de asiento

4.4.3.2. Consultas dependientes

Tal y como hemos explicado en el anterior objeto de negocio que son las consultas dependientes, este tambien tiene sus propias consultas para poder dar funcionalidades al formulario.

Nuestro objeto de negocio (pf_cg_factura) usará las siguientes consultas dependientes:

mdqo_divisa_c

Consulta:	pf_divisacont_doc		
Dependencias:	Campo	Tipo	Valor
	heentite	Control	txtentidad_id
	hecoddev	Control	txtdivisa_id

mdqo_cuenta_c

Consulta:	cpt_cp_gct_vdl		
Dependencias:	Campo	Tipo	Valor
	cpcompte	Control	txtcuenta
	txtplan	Control	cpnumpc

mdqo_cuenta_d

Consulta:	cpt_cp_gct_vdl		
Dependencias:	Campo	Tipo	Valor
	txtdescripcion	Control	cpintico
	txtcuenta	Control	cpcompte
	txtplan	Control	cpnumpc

mdqo_cif_nif_c

Consulta:	cpt_cp_gct_vdl		
Dependencias:	Campo	Tipo	Valor
	txtcif_nif	Control	cpsiren
	cpcompte	Control	txtcuenta

mdqo_codigo_iva_c

Consulta:	cpt_taxes_vdl		
Dependencias:	Campo	Tipo	Valor
	txtcodigo_iva	Control	txcodtax
	txtplan	Control	txnumpc

mdqo_codigo_iva_d

Consulta:	cp_taxes_vdl_d		
Dependencias:	Campo	Tipo	Valor
	txtcodigo_iva	Control	txcodtax
	txtplan	Control	txnumpc

mdqo_tipo_impositivo

Consulta:	fi_modalidades_vdl		
Dependencias:	Campo	Tipo	Valor
	txtcodigo_iva	Control	txcodtax
	txttipo_impositivo	Control	txtax
	txt_tipo_impositivo	Control	xtipo_impositivo

4.4.3.3. Formulario Entrada Factura < pf_cg_factura >

La siguiente captura corresponde al formulario de **Entrada automática**:

Entrada automática

Lectura de importes:
Primero:
Segundo:
Tercero:
Cuarto:

Entrada de datos:

Multi-IVA: ☐

Datos Factura:
Núm. factura: Cuenta:
Fecha factura: Descripción:
Fecha contable: CIF/NIF:
Divisa: Total factura:

Lectura de importes (IVA) | IRPF | Vencimientos | Contrapartida (Gastos/Ingresos)

Cód.	IVA	Mod.	Tp. impt.	Base	% IVA	Cuota	% Rec.	Cuota Rec.	Cuenta	Cuenta R.

Asientos generados:

Diario	Nº Asiento	Tipo doc.	F. contable	Núm. Doc.	Total

Figura 22: Entrada Automática

Utilizando los botones del formulario podremos ver el resto de ventanas del formulario:

-IVA:

Lectura de importes (IVA) | IRPF | Vencimientos | Contrapartida (Gastos/Ingresos)

Cód.	IVA	Mod.	Tp. impt.	Base	% IVA	Cuota	% Rec.	Cuota Rec.	Cuenta	Cuenta Recargo

Figura 23: Entrada Automática: Desglose IVA

-IRPF:

Lectura de importes (IVA)IRPFVencimientosContrapartida (Gastos/Ingresos)

	Retención	Base	%	Importe	Cuenta

Figura 24: Entrada Automática: Desglose IRPF

-Vencimientos:

Lectura de importes (IVA)IRPFVencimientosContrapartida (Gastos/Ingresos)

	Modo pago		F. vencimiento	Importe

Figura 25: Entrada Automática: Desglose Vencimientos

-Contrapartida (Gastos/Ingresos):

Lectura de importes (IVA)IRPFVencimientosContrapartida (Gastos/Ingresos)

	Libro	Cuenta		Importe

Desg. analítico

Figura 26: Entrada Automática: Desglose Contrapartida

-Desglose analítico:

Desglose analítico

	Libro	Cuenta		Importe

Figura 27: Entrada Automática: Desglose Analítico

-Ver asiento:

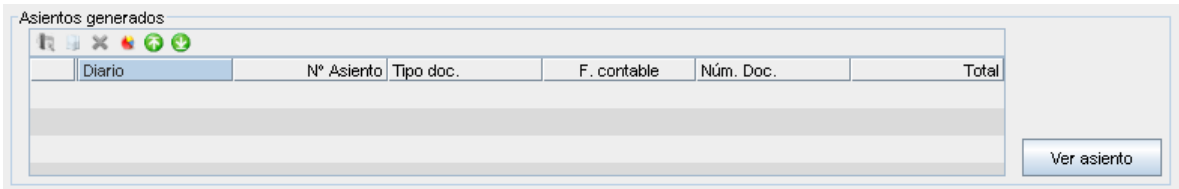


Figura 28: Entrada Automática: Asientos generados

4.5. Plan de Pruebas

Para la realización de las pruebas utilizaremos la técnica de pruebas basadas en casos de uso, con lo cual veremos si el sistema cumple con las funcionalidades, así como si está bien integrado con los demás componentes con los que interactúa, siguiendo el flujo básico, así como también a que responda correctamente a los diferentes flujos alternativos que pudieran suceder durante la ejecución de los procesos.

4.5.1. Prueba 1

A continuación se mostrará una tabla con las pruebas mínimas a realizar para comprobar la correcta funcionalidad del sistema siguiendo la “Lectura de Datos básica” que hará siempre que queramos leer una factura con el lector.

Escenario	Condición	Resultado esperado
Escenario 1	Seleccionar Botón “Inicio Entrada”	En la sección de entrada de Datos tiene que aparecer el nombre del control que se va a leer 1ro antes de los importes seleccionados, concretamente “Número Factura”
Escenario 2	Escenario 1 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 1) +Siguiente nombre de datos a leer, “Fecha Factura”
Escenario 3	Escenario 2 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 2) + Si fecha contable no era indicada en el formulario de datos previos, se le asignar la misma que el valor leído como fecha factura + Siguiente nombre de datos a leer, “NIF/Descripción”
Escenario 4	Escenario 3 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 3) + Partiendo del nombre Completo del representante de la cuenta o de su NIF Cargar los datos de número de cuenta, descripción o CIF/NIF + Siguiente nombre de datos a leer Primer Importe Seleccionado

4.5.2. Prueba 2

A continuación se mostrará una tabla con las pruebas mínimas a realizar para comprobar la correcta funcionalidad del sistema siguiendo la Entrada informando 4 importes (a leer con el lector) y sin multi-IVA

Escenario	Condición	Resultado esperado
Escenario 1	Selección de orden de lectura de importes con el siguiente orden: Base, IVA, Recargo, Total	Selección correcta de valores
Escenario 2	En este caso no seleccionar	

	check de Multi IVA	
Escenario 3	Lectura de Datos básica + pulsar tabulador	Siguiente nombre de datos a leer Primer Importe Seleccionado
Escenario 4	Escenario 3 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 3) + Cargar libro y cuenta Contrapartida partiendo de la cuenta + Siguiente nombre de datos a leer Segundo Importe Seleccionado
Escenario 5	Escenario 4 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 4) + Siguiente nombre de datos a leer Tercero Importe Seleccionado
Escenario 6	Escenario 5 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 5) +Calcular porcentaje de IVA a partir de la Base y de la cuota+ Calcular porcentaje de Recargo a partir de Base y de la cuota de recargo+ Calcular Total Línea IVA sumando Base, Cuota y Cuota Recargo, a partir de los porcentajes obtener el tipo impositivo+ Partiendo del tipo impositivo obtener la modalidad y el Código de IVA y asignarlos aunque no concuerden con los de la cuenta+ Siguiente nombre de datos a leer Cuarto Importe Seleccionado
Escenario 7	Escenario 6 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 6) +Calcular Retención + Calcular Vencimientos + Calcular Contrapartida + Finalizar entrada datos con el lector

4.5.3. Prueba 3

A continuación se mostrará una tabla con las pruebas mínimas a realizar para comprobar la correcta funcionalidad del sistema siguiendo la Entrada informando 3 importes a leer con el lector que seran (Base, IVA y Recargo) y sin multi-IVA

Escenario	Condición	Resultado esperado
Escenario 1	Selección de orden de lectura de importes con el siguiente orden: Base, IVA, Recargo	Selección correcta de valores
Escenario 2	En este caso no seleccionar check de Multi IVA	
Escenario 3	Lectura de Datos básica + pulsar tabulador	Siguiente nombre de datos a leer Primer Importe Seleccionado
Escenario 4	Escenario 3 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 3) + Cargar libro y cuenta Contrapartida partiendo de la cuenta + Siguiente nombre de datos a leer Segundo Importe Seleccionado
Escenario 5	Escenario 4 + Leer datos solicitados + pulsar	R.E.(Escenario 4) + Siguiente nombre de datos a leer Tercero Importe Seleccionado

	tabulador	
Escenario 6	Escenario 5 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 5) + Calcular importes + Calcular Retención + Calcular Vencimientos + Calcular Contrapartida + Finalizar entrada datos con el lector

4.5.4. Prueba 4

A continuación se mostrará una tabla con las pruebas mínimas a realizar para comprobar la correcta funcionalidad del sistema siguiendo la Entrada informando 1 importe a leer con el lector.

Escenario	Condición	Resultado esperado
Escenario 1	Selección de orden de lectura de importes con el siguiente orden: Total	Selección correcta de valores
Escenario 2	En este caso no seleccionar check de Multi_IVA	Independientemente de los importes a leer seleccionado, obligatoriamente nos preguntara como importes el Total.
Escenario 3	Lectura de Datos básica + pulsar tabulador	Siguiente nombre de datos a leer Primer Importe Seleccionado
Escenario 4	Escenario 3 + Leer datos solicitados + pulsar tabulador	R.E. (Escenario 5) +Calcular Retención + Calcular Vencimientos + Calcular importe Contrapartida+ sale de la entrada de datos con el lector.

4.5.5. Prueba 5

A continuación se mostrará una tabla con las pruebas mínimas a realizar para comprobar la correcta funcionalidad del sistema siguiendo la Entrada de Datos con el lector con multi-IVA.

Escenario	Condición	Resultado esperado
Escenario 1	Selección de orden de lectura de importes con el siguiente orden: Base, IVA	Selección correcta de valores
Escenario 2	En este caso seleccionar check de Multi_IVA	Al seleccionar-lo, independientemente de los importes a leer seleccionado, obligatoriamente nos preguntara como importes la base y el IVA.
Escenario 3	Lectura de Datos básica + pulsar tabulador	Siguiente nombre de datos a leer Primer Importe Seleccionado
Escenario 4	Escenario 3 + Leer datos solicitados + pulsar tabulador	R.E.(Escenario 3) + Cargar libro y cuenta Contrapartida partiendo de la cuenta + Siguiente nombre de datos a leer Segundo Importe Seleccionado

Escenario 5	Escenario 4 + Leer datos solicitados + pulsar tabulador	<p>Si no se pulsa final Entrada:</p> <p>R.E.(Escenario 5) +Calcular Retención + Calcular Vencimientos + Calcular importe Contrapartida + Volver a leer 1r nombre de importe y seguir al escenario 4 si no se pulsa final Entrada</p> <p>Si se pulsa final de IVA:</p> <p>R.E. (Escenario 5) +Calcular Retención + Calcular Vencimientos + Calcular importe Contrapartida+ sale de la entrada de datos con el lector.</p>
-------------	---	--

5. Construcción

5.1. Tecnología utilizada

Durante la realización del proyecto se han utilizado varias tecnologías. Para la implementación de las tablas que creamos y las tablas a las que realizamos consultas se utiliza SQL, para la codificación se ha utilizado Java ya que se trata de un lenguaje de programación orientada a objetos que permite mucha más flexibilidad a la hora de programar que C y C++, la programación del cual se ha hecho mediante el entorno de desarrollo Eclipse Galileo. En cuanto a los formularios de la aplicación se han desarrollado en Ottools 7.5.

5.1.1. Lenguajes de programación

Java 1.6

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

SQL

SQL (Structured Query Language) aparece en 1974 cuando Donald Chamberlin definió un lenguaje para la especificación de las características de las bases de datos que adoptaban el modelo relacional. Este lenguaje se llamaba SEQUEL, pero entre 1976 y 1977 se cambió de nombre por motivos legales y fue llamado SQL. Sus características principales son:

- Es un lenguaje de alto nivel o de no procedimiento, que gracias a su fuerte base teórica y su orientación al manejo de conjunto de registros, permite una alta productividad en la codificación.
- Asume las siguientes funcionalidades:
 - Lenguaje de definición de datos.
 - Lenguaje de definición de vistas.
 - Lenguaje de manipulación de datos.
 - Permite concesión y denegación de permisos.
 - Implementa restricciones de integridad y controles de transacción.

5.1.2. Entorno de desarrollo

Ottools 7.5

Ottools es un completo entorno de desarrollo de aplicaciones, las cuales se realizan de manera rápida y sencilla. Es una tecnología totalmente orientada a objetos, bajo una estructura de cliente/servidor. Incluye un potente sistema que permite la definición de consultas mediante la combinación de tablas y enlaces, distintos criterios de selección y ordenación.

Con un mismo desarrollo, Ottools permite la ejecución de la aplicación tanto en entorno cliente/servidor como en entorno web.

Eclipse Galileo

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las

aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

5.1.3. Estilo de Codificación

En la implementación de la aplicación, tanto del código como en la definición de tablas, campos, etc. se han seguido unas pautas de codificación que se especifican a continuación:

- Los nombres de las variables empiezan en minúscula.
- El nombre de las variables y funciones indican su rol desempeñado.
- En el caso de que un nombre de variable o función esté formado por dos o mas palabras, la primera letra de cada palabra se pone en mayúscula, excepto si se trata de la palabra inicial
- Objeto de negocio:
 - Si la variable es un check box su nombre empezará por “chk”.
 - Si la variable es texto o valores numéricos su nombre empezará por “txt”.
 - Si la variable es un integer su nombre empezará por “cbo”.
- Tabla de base de datos:
 - Los nombres de los campos empezarán por una “x”.
- Las listas del sistema empezarán por “syf_”.
- El nombre de las consultas a la base de datos empezará por “mdqo_”.

5.2. Normas

Para poder seguir unos estándares y poder crear un software de calidad y que permita que cualquier persona de la empresa pueda realizar modificaciones en las aplicaciones echas sin necesidad de tener que haber creado la aplicación, se han creado una normativa a emplear a la hora de hacer una aplicación la cual se explica a continuación.

5.2.1. Formularios

- **Máscaras.** Las máscaras output relativas a campos numéricos con decimales, se definirán incluyendo un cero en la posición de las unidades, y tantos ceros como posiciones se hayan definido.
- **Check Box.** Los checkbox siempre estarán Enable, y se activarán o no, modificando la propiedad ReadOnly, ya sea mediante un Dynamic Status (GFD) o Cmaint.ControlReadOnly.
- **Justificación.** Los contenedores de valores numéricos tienen que tener la propiedad de *TextAlign* informada con el valor *Right Justify* y en las fechas el valor de esta propiedad tiene que ser Center.
- **Tamaño máximo.** El tamaño máximo de un formulario, sin que aparezca la barra de desplazamiento vertical u horizontal, es:

	HEIGHT	WIDTH
800 x 600	7265	11940
1024 x 768	9472	15307

- Los nombres de los botones del formulario empezarán por “cmd”
- Los nombres de los grid del formulario empezarán por “grd_”

5.2.2. Entorno Eclipse

- Las variables tienen que empezar en minúscula.
- Los nombres de las funciones tienen que explicar lo que hacen.
- El package que se creará se tiene que llamar “com.ccsagrosso.gen.finanzas.lector”
- Las clases que se creen tienen que tener el mismo nombre que el formulario que las utiliza.
- Las funciones para realizar los cálculos tienen que estar dentro de una clase con el nombre del formulario que las utiliza y el nombre de la clase que las contiene tiene que terminar con “_general”.

6. Pruebas

Para comprobar el correcto funcionamiento del sistema se realizaron pruebas unitarias descritas en el plan de pruebas especificado en el documento de análisis y diseño.

Al haber hecho las pruebas a distintos niveles, es decir, desde un nivel más físico a uno más lógico (o de cumplimiento de funcionalidades), se puede decir que se realizó un testeo de la

aplicación bastante exhaustivo, a la búsqueda de posibles errores, los cuales fueron apareciendo y fueron corregidos.

6.1. Pruebas unitarias

Se entiende por pruebas unitarias las pruebas que se realizan para comprobar el correcto funcionamiento de un módulo de código. Esto sirve para probar la corrección de los módulos por separado. En este proyecto, se realizaron dichas pruebas durante la programación de la aplicación, durante la fase de diseño y análisis, en el punto 4.5 de la memoria podemos verlo con más detalle.

6.1.1. Prueba 1

En este módulo de primera aplicación se fueron realizando las pruebas a medida que se iban creando funciones. Hasta no lograr el correcto funcionamiento de la función y también el correcto funcionamiento entre funciones que pudieran llamar a la que se estaba probando, no se pasaba a programar la siguiente; por lo que se realizaron pruebas exhaustivas de código. A continuación tenemos la lista de pruebas realizadas:

- Comprobación del correcto funcionamiento de los algoritmos que insertan la información en los campos correctos partiendo del dato pedido.
- Validación correcta de los datos insertados en los campos.
- Comprobación del correcto funcionamiento de la función que asigna la fecha factura como fecha contable y de la función que, partiendo del NIF o descripción, asigna una cuenta bancaria.

6.1.2. Prueba 2

Como en el módulo anterior, se hicieron las mismas pruebas iniciales ya que antes de pedir los valores se insertan los datos del modelo anterior. Seguidamente se muestra la lista de pruebas realizadas a nivel de funciones:

- Comprobación del correcto funcionamiento de los algoritmos que insertan la información en los campos correctos partiendo del dato pedido.
- Asignación correcta de los nombres de los importes que se pedían con el orden correcto hasta llegar al cuarto importe y correcta finalización de la entrada de datos.
- Comprobación del correcto funcionamiento de las funciones que calculan los porcentajes de IVA y que asignan códigos de IVA y calculan retenciones, contrapartidas y vencimientos.

6.1.3. Prueba 3

Como en el primer módulo, se hicieron las mismas pruebas iniciales ya que antes de pedir los valores se insertan los datos del modelo anterior. Seguidamente se muestra la lista de pruebas realizadas a nivel de funciones:

- Comprobación del correcto funcionamiento de los algoritmos que insertan la información en los campos correctos partiendo del dato pedido.
- Asignación correcta de los nombres de los importes que se pedían con el orden correcto hasta llegar al tercer importe y correcta finalización de la entrada de datos.
- Comprobación del correcto funcionamiento de las funciones que calculan los porcentajes de IVA y que asignan códigos de IVA y calculan retenciones, contrapartidas y vencimientos.

6.1.4. Prueba 4

Como en el primer módulo, se hicieron las mismas pruebas iniciales ya que antes de pedir los valores se insertan los datos del modelo anterior. Seguidamente se muestra la lista de pruebas realizadas a nivel de funciones:

- Comprobación del correcto funcionamiento de los algoritmos que insertan la información en los campos correctos partiendo del dato pedido.
- Asignación correcta del nombre de Total como importe que pedía y correcta finalización de la entrada de datos.
- Comprobación del correcto funcionamiento de las funciones que calculan los porcentajes de IVA y que asignaban códigos de IVA y calculan retenciones, contrapartidas y vencimientos.

6.1.5. Prueba 5

Como en el primer módulo, se hicieron las mismas pruebas iniciales ya que antes de pedir los valores se insertan los datos del modelo anterior. Seguidamente se muestra la lista de pruebas realizadas a nivel de funciones:

- Comprobación del correcto funcionamiento de los algoritmos que insertan la información en los campos correctos partiendo del dato pedido.
- Asignación correcta de los nombres de Base e IVA a los importes y correcto funcionamiento al volver a pedir los valores hasta pulsar la finalización de la entrada de datos.
- Comprobación del correcto funcionamiento de las funciones que calculan los porcentajes de IVA y que asignan códigos de IVA y calculan retenciones, contrapartidas y vencimientos.

7. Conclusiones

7.1. Objetivos cumplidos

Los principales objetivos marcados durante la fase de definición de requerimientos, se han cumplido con éxito, aunque la complejidad de ciertos aspectos de la aplicación hizo que se demorara el tiempo más de lo que se planificó en un inicio. A continuación se enumeran los objetivos conseguidos:

- Crear asientos contables a partir de la lectura con el lector
- Informar manualmente los campos que no hayan sido informados o se quieran modificar
- Agilizar la entrada de asientos
- Una vez entrados los datos necesarios, se puede modificar el borrador, validarlo o contabilizarlo para poder ser gestionado por ekon finanzas.

7.2. Líneas de trabajo abiertas

Dentro de las líneas de trabajo abiertas se puede destacar el echo de tener en cuenta el tratamiento de la cuota de deducción fiscal la cual en la actualidad no se trata y otra mejora sería el echo de poder guardar los tipos de importe que se leen por cada tipo de factura y poder cargar estos valores automaticamente una vez se entre una factura del mismo tipo.

7.3. Valoración personal

En la realización del proyecto he podido poner en práctica los conocimientos adquiridos durante la carrera y también he podido aprender un nuevo entorno de desarrollo como es Ottools, así como el proceso que se sigue a la hora de realizar una aplicación.

He podido ver como es la dinámica de trabajo de una empresa de desarrollo de software, ya que nunca había trabajado en una.

Respecto al proyecto en sí, al principio me resulto un poco complejo familiarizarme con los conceptos de finanzas y en cómo obtener y calcular ciertos valores y como retornar ciertos datos. Aunque me adapte bien y a medida que pasaba el tiempo adquiría mayor conocimiento y al final pude realizarlo todo sin demasiadas complicaciones.

El proyecto me ha servido tanto para ganar experiencia en el mundo de las empresas de desarrollo de software como para entregar el proyecto de final de carrera, por lo que puedo decir que la realización del proyecto me ha resultado satisfactoria.

Bibliografía
Fuentes

- **Berthold Daum , “Eclipse 3 para desarrolladores Java”. Primera edición. Ed. Anaya Multimedia, 2005.**
 - Libro en el que explica la base sobre java y programar con eclipse.
- **LUIS JOYANES AGUILAR, “ESTRUCTURAS DE DATOS EN JAVA”. Primera Edición. S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, 2007.**
 - Libro en el que explica las estructuras de datos en java y la teoría de la programación en java.
- **BRUCE ECKEL, “PIENSA EN JAVA”, Primera edición, PRENTICE-HALL, 2007.**
 - Libro en el que explica cómo se programa en java y la teoría de la programación en java.
- **Normas de desarrollo**
 - Ayuda referente a las normas internas de diseño y programación de las aplicaciones realizadas dentro de UNIT 4.
- **Sistema de la Calidad de UNIT 4**
 - Manual en el que se explican las abreviaturas a utilizar en la declaración de controles, etc., así como todo lo relacionado con la calidad del software.

Fuentes electrónicas

- <http://www.w3schools.com/sql/default.asp>
 - Web muy completa en la cual hay ejemplos y se explica cómo funciona el lenguaje SQL.
 - Fecha último acceso 23/06/2010
- <http://es.wikipedia.org/wiki/Portada>
 - Enciclopedia online.
 - Fecha último acceso 23/06/2010
- <http://todojava.awardspace.com/>
 - Web con ejemplos de Java.
 - Fecha último acceso: 23/06/2010

Lunes 28 de Junio de 2010

Jordi Malé Marbà